

## AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

### Listing of Claims:

1           1. (Original) A method for mapping data in a markup language document  
2 to an object model, the method comprising the steps of;  
3           receiving a mapping request for mapping data in a markup language  
4 document having data architecture into an object model; and  
5           mapping, in response to the mapping request, the data into the object  
6 model using mapping meta-data which defines how the data architecture of the  
7 markup language document maps to the object model.

1           2. (Original) The method as claimed in claim 1, wherein the mapping  
2 request includes a key for identifying the markup language document and the  
3 mapping step obtains the markup language document using the key.

1           3. (Original) The method as claimed in claim 1, wherein the markup  
2 language document has one or more elements containing data, the object model  
3 has one or more object classes, each object class has one or more attributes that  
4 correspond to the elements, and the step of mapping includes a step of populating  
5 the attributes with the data of the corresponding elements based on the mapping  
6 meta-data.

1           4. (Original) The method as claimed in claim 1, wherein the markup  
2 language document has one or more elements containing data, the object model

3 has one or more object classes, each object class has one or more attributes that  
4 correspond to the elements and the step of mapping includes;  
5 a step of generating a row structure corresponding to the markup language  
6 elements of the markup language document;  
7 a step of converting the row structure into one or more objects  
8 corresponding to the elements; and  
9 a step of populating attributes of the converted objects with the data of the  
10 elements based on the mapping meta-data.

1 5. (Original) The method as claimed in claim 3, wherein the markup  
2 language document further has at least one element containing one or more other  
3 elements and the mapping step inserts, based on the mapping meta-data, a value  
4 representing the relation between the at least one element and the one or more  
5 other elements into an attribute of the object model to represent a relationship  
6 between objects corresponding to the at least one element and the one or more  
7 other elements.

1 6. (Original) The method as claimed in claim 5, wherein the at least one  
2 element contains a single element containing data and the mapping step inserts a  
3 value representing the relation between the at least one element and the single  
4 element into an attribute of the object model that represents a one-to-one  
5 relationship between objects that correspond to the at least one element and the  
6 single element.

1 7. (Original) The method as claimed in claim 5, wherein the at least one  
2 element contains a single element containing a pointer to another element in a  
3 different markup language document and the mapping step inserts a value  
4 representing the relation between the at least one element and the single element

5 into an attribute of the object model that represents an aggregate one-to-one  
6 relationship between objects that correspond to the at least one element and the  
7 single element.

1 8. (Original) The method as claimed in claim 5, wherein the at least one  
2 element contains multiple elements containing data and the mapping step inserts  
3 values representing the relation between the at least one element and the multiple  
4 elements into attributes of the object model that represent one-to-many  
5 relationships between objects that correspond to the at least one element and the  
6 multiple elements.

1 9. (Original) The method as claimed in claim 5, wherein the at least one  
2 element contains multiple elements containing pointers to elements in one or more  
3 different markup language documents and the mapping step inserts values  
4 representing the relation between the at least one element and the multiple  
5 elements into attributes of the object model that represent aggregate one-to-many  
6 relationships between objects that correspond to the at least one element and the  
7 multiple elements.

1 10. (Original) The method as claimed in claim 1 further comprising a step  
2 of obtaining the mapping meta-data prior to the mapping step.

1 11. (Currently amended) The method as claimed in ~~claim 1~~ claim 10,  
2 wherein the obtaining step is carried out during initialization of a system for  
3 executing the receiving step and the mapping step.

1 12. (Original) The method as claimed in claim 1, wherein the markup  
2 language document has one or more elements, the object model has one or more

3 object classes, and the mapping meta-data includes mapping information  
4 regarding one of the elements and the corresponding object class.

1 13. (Original) The method as claimed in claim 1, wherein the markup  
2 language document has one or more elements, the object model has one or more  
3 object classes, each object class has one or more attributes, the mapping meta-data  
4 includes mapping information regarding one of the elements that contains data  
5 and the corresponding attribute, and the mapping step maps the data of the one of  
6 the elements into the corresponding attributes based on the mapping information.

1 14. (Original) The method as claimed in claim 1, wherein the markup  
2 language document is a document in which each element is defined by indicators.

1 15. (Original) The method as claimed in claim 14, wherein the markup  
2 language document is an eXtensible Markup Language (XML) document.

1 16. (Original) A method for mapping an object in an object model to a  
2 markup language document, the method comprising the steps of;  
3 receiving a mapping request for mapping one or more objects of an object  
4 model into a markup language document having data architecture; and  
5 mapping, in response to the mapping request, the objects into the markup  
6 language document using mapping meta-data which defines how the object model  
7 maps to the data architecture of the markup language document.

1 17. (Original) The method as claimed in claim 16, wherein the mapping  
2 request includes a key for identifying one of the objects and the mapping step  
3 includes a step of locating the markup language document where said object is to  
4 be inserted.

1 18. (Original) The method as claimed in claim 16, wherein the object  
2 model has one or more object classes containing the one or more objects to be  
3 mapped, each object class has one or more attributes, and the step of the mapping  
4 includes a step of creating one or more elements of the markup language  
5 document corresponding to the one or more objects by inserting values of the  
6 attributes into the elements based on the mapping meta-data.

1 19. (Original) The method as claimed in claim 16, wherein the object  
2 model has one or more object classes containing one or more objects to be  
3 mapped, each object class has one or more attributes, and the step of mapping  
4 includes:  
5 a step of generating a row structure corresponding to the one or more  
6 objects based on the mapping meta-data; and  
7 a step of creating elements of the markup language document based on the  
8 row structure.

1 20. (Original) The method as claimed in claim 18, wherein the attributes  
2 include an attribute representing a relationship between the objects and the  
3 mapping step maps a value representing the relationship between the elements.

1 21. (Original) The method as claimed in claim 20, wherein an attribute  
2 represents a one-to-one relationship between a source object and a target object  
3 and the mapping step maps a value representing the one-to-one relationship to an  
4 element containing another element for containing data that corresponds to data of  
5 the target object.

1 22. (Original) The method as claimed in claim 20, wherein an attribute  
2 represents an aggregate one-to-one relationship between a source object and a

3 target object and the mapping step maps a value representing the aggregate one-  
4 to-one relationship to an element containing another element for containing a  
5 pointer to point to another element in a different markup language document that  
6 contains data corresponding to data of the target object.

1 23. (Original) The method as claimed in claim 20, wherein an attribute  
2 represents a one-to-many relationship between a source object and multiple target  
3 objects and the mapping step maps values representing the one-to-many  
4 relationship to an element containing multiple other elements for containing data  
5 that correspond to data of the multiple target objects.

1 24. (Original) The method as claimed in claim 20, wherein an attribute  
2 represents an aggregate one-to-many relationship between a source object and  
3 multiple target objects and the mapping step maps values representing the  
4 aggregate one-to-many relationship to an element containing multiple other  
5 elements for containing pointers to points other elements in one or more different  
6 markup language documents that contain data corresponding to data of the target  
7 objects.

1 25. (Original) The method as claimed in claim 16 further comprising a  
2 step of obtaining the mapping meta-data prior to the mapping step.

1 26. (Currently amended) The method as claimed in ~~claim 16~~ claim 25,  
2 wherein the obtaining step is carried out during initialization of a system for  
3 executing the receiving step and the mapping step.

1 27. (Original) The method as claimed in claim 16, wherein the markup  
2 language document has one or more elements, the object model has one or more

3 object classes, and the mapping meta-data includes information regarding the  
4 object class and the corresponding one of the elements.

1 28. (Original) The method as claimed in claim 16, wherein the markup  
2 language document has one or more elements, the object model has one or more  
3 object classes, the object class has one or more attributes, and the mapping meta-  
4 data includes information regarding one of the attributes and the corresponding  
5 one of the elements.

1 29. (Original) The method as claimed in claim 16, wherein the markup  
2 language document is a document in which each element is defined by indicators.

1 30. (Original) The method as claimed in claim 29, wherein the markup  
2 language document is a XML document.

1 31. (Original) A mapping manager for mapping between a markup  
2 language document and an object model, the mapping manager comprising:  
3 an executor for receiving a mapping request for mapping between a  
4 markup language document having data architecture and an object model; and  
5 a mapping executor for mapping, in response to the mapping request,  
6 between data of the markup language document and objects of the object model  
7 using mapping meta-data which defines how the data architecture of the markup  
8 language document maps to the object model.

1 32. (Original) The manager as claimed in claim 31, wherein the mapping  
2 request includes a key for identifying the markup language document and the  
3 mapping executor includes an accessor to obtain the markup language document  
4 using the key.

1           33. (Original) The manager as claimed in claim 31, wherein the markup  
2 language document has one or more elements, the object model has one or more  
3 object classes, each object class has one or more attributes, and the mapping  
4 executor includes a mapping unit for populating the attributes with the data of the  
5 elements based on the mapping meta-data.

1           34. (Original) The manager as claimed in claim 31, wherein the markup  
2 language document has one or more elements, the object model has one or more  
3 object classes, each object class has one or more attributes, and the mapping  
4 executor includes:

5           a generator for generating a row structure corresponding to the markup  
6 language elements;

7           a converter for converting one or more objects based on the row structure;  
8 and

9           a mapping unit for populating attributes of the converted objects with the  
10 data of the elements based on the mapping meta-data.

1           35. (Original) The manager as claimed in claim 33, wherein the markup  
2 language document further has at least one element containing one or more other  
3 elements, and the mapping unit inserts, based on the mapping meta-data, a value  
4 representing the relation between the at least one element and the one or more  
5 other elements into an attribute of the object model to represent a relationship  
6 between objects corresponding to the at least one element and the one or more  
7 other elements.

1           36. (Original) The manager as claimed in claim 35, wherein the at least  
2 one element contains a single element containing data and the mapping unit  
3 inserts a value representing a relation between the at least one element and the



4 single element into an attribute of the object model that represents a one-to-one  
5 relationship between objects that corresponds to the at least one element and the  
6 single element.

1 37. (Original) The manager as claimed in claim 35, wherein the at least  
2 one element contains a single element containing a pointer to another element in a  
3 different markup language document, and the mapping unit inserts a value  
4 representing the relation between the at least one element and the single element  
5 into an attribute of the object model that represents an aggregate one-to-one  
6 relationship between objects that corresponds to the at least one element and the  
7 single element.

1 38. (Original) The manager as claimed in claim 35, wherein the at least  
2 one element contains multiple elements containing data, and the mapping unit  
3 inserts values representing the relation between the at least one element and the  
4 multiple elements into attributes of the object model that represent one-to-many  
5 relationships between objects that corresponds to the at least one element and the  
6 multiple elements.

1 39. (Original) The manager as claimed in claim 35, wherein the at least  
2 one element contains multiple elements containing pointers to elements in one or  
3 more different markup language documents, and the mapping unit inserts values  
4 representing the relation between the at least one element and the multiple  
5 elements into attributes of the object model that represent aggregate one-to-many  
6 relationships between objects that correspond to the at least one element and the  
7 multiple elements.

1           40. (Original) The manager as claimed in claim 31, wherein the mapping  
2 executor includes a mapping unit for obtaining the mapping meta-data.

1           41. (Original) The manager as claimed in claim 31, wherein the markup  
2 language document has one or more elements, the object model has one or more  
3 object classes, and the mapping executor includes a mapping unit for handling a  
4 mapping between one of the elements and the corresponding object class.

1           42. (Original) The manager as claimed in claim 31, wherein the markup  
2 language document has one or more elements, the object model has one or more  
3 object classes, each object class has one or more attributes, and the mapping  
4 executor includes a mapping unit for handling a mapping between one of the  
5 elements and the corresponding attribute.

1           43. (Original) The manager as claimed in claim 31, wherein the markup  
2 language document has one or more elements, the object model has one or more  
3 object classes, each object class has one or more attributes, the attributes include  
4 an attribute representing a relationship between the objects, the mapping executor  
5 includes a relationship mapping unit for handling a mapping of a relationship  
6 between the objects, and the relationship represents a relation between the  
7 elements.

1           44. (Original) The manager as claimed in claim 31, wherein the object  
2 model has one or more object classes, each object class has one or more attributes,  
3 and the mapping executor includes a mapping unit for creating one or more  
4 elements corresponding to the attributes by inserting values of the attributes based  
5 on the mapping meta-data.

1           45. (Original) The manager as claimed in claim 31, wherein the markup  
2 language document is a document in which each element is defined by indicators.

1           46. (Original) The manager as claimed in claim 45, wherein the markup  
2 language document is an XML document.

1           47. (Original) A mapping system for mapping between a markup language  
2 document and an object model, the mapping system comprising:

3           an executor for receiving a mapping request for mapping between a  
4 markup language document having data architecture and an object model;  
5           a storage for storing mapping meta-data which defines how the data  
6 architecture of the markup language document maps to the object model; and  
7           a mapping executor for mapping, in response to the mapping request,  
8 between data of the markup language document and an object of the object model  
9 using the mapping meta-data.

1           48. (Original) The system as claimed in claim 47 wherein the mapping  
2 executor includes a mapping unit for obtaining the mapping meta-data from the  
3 storage.

1           49. (Original) The system as claimed in claim 47, wherein the mapping  
2 storage obtains the mapping meta-data prior to an operation of the mapping  
3 executor.

1           50. (Original) The system as claimed in claim 47, wherein the mapping  
2 storage obtains the mapping meta-data during initialization of the system.

1           51. (Original) The system as claimed in claim 47 further comprising a  
2 runtime interface to accept the mapping request from an application.

1           52. (Original) The system as claimed in claim 47, wherein the markup  
2 language document has one or more elements, the object model has one or more  
3 object classes, the object class has one or more attributes, and the mapping  
4 executor includes a mapping unit for populating the attributes with the data  
5 associated with the elements based on the mapping meta-data.

1           53. (Original) The system as claimed in claim 47, wherein the object  
2 model has one or more object classes, each object class has one or more attributes,  
3 and the mapping executor includes a mapping unit for creating one or more  
4 elements corresponding to the attributes by inserting values of the attributes based  
5 on the mapping meta-data.

1           54. (Original) The system as claimed in claim 47, wherein the markup  
2 language document is a document in which each element is defined by indicators.

1           55. (Original) The system as claimed in claim 54, wherein the markup  
2 language document is an XML document.

1           56. (Original) Computer readable media storing the instructions or  
2 statements for use in the execution in a computer of a method for mapping data in  
3 a markup language document to an object model, the method comprising the steps  
4 of;  
5           receiving a mapping request for mapping data in a markup language  
6 document having data architecture into an object model; and

7 mapping, in response to the mapping request, the data into the object  
8 model using mapping meta-data which defines how the data architecture of the  
9 markup language document maps to the object model.

1 57. (Original) Electronic signals for use in the execution in a computer of a  
2 method for mapping data in a markup language document to an object model, the  
3 method comprising the steps of;

4 receiving a mapping request for mapping data in a markup language  
5 document having data architecture into an object model; and

6 mapping, in response to the mapping request, the data into the object  
7 model using mapping meta-data which defines how the data architecture of the  
8 markup language document maps to the object model.

1 58. (Original) A computer program product for use in the execution in a  
2 computer of a method for mapping data of a markup language document to an  
3 object model, the computer program product comprising:

4 a module for receiving a mapping request for mapping data in a markup  
5 language document having data architecture into an object model; and

6 a module for mapping, in response to the mapping request, the data into  
7 the object model using mapping meta-data which defines how the data  
8 architecture of the markup language document maps to the object model.

1 59. (Original) Computer readable media storing the instructions or  
2 statements for use in the execution in a computer of a method for mapping an  
3 object in an object model to a markup language document, the method comprising  
4 the steps of:

5 receiving a mapping request for mapping one or more objects of an object  
6 model into a markup language document having data architecture; and

7 mapping, in response to the mapping request, the objects into the markup  
8 language document using mapping meta-data which defines how the object  
9 model maps to the data architecture of the markup language document.

1 60. (Original) Electronic signals for use in the execution in a computer of a  
2 method for mapping an object in an object model to a markup language document,  
3 the method comprising the steps of:

4 receiving a mapping request for mapping one or more objects of an object  
5 model into a markup language document having data architecture; and

6 mapping, in response to the mapping request, the objects into the markup  
7 language document using mapping meta-data which defines how the object model  
8 maps to the data architecture of the markup language document..

1 61. (Original) A computer program product for use in the execution in a  
2 computer of a method for mapping an object in an object model to a markup  
3 language document, the computer program product comprising:

4 a module for receiving a mapping request for mapping one or more objects  
5 of an object model into a markup language document having data architecture;  
6 and

7 a module for mapping, in response to the mapping request, the objects into  
8 the markup language document using mapping meta-data which defines how the  
9 object model maps to the data architecture of the markup language document.